



Distance Learning System

MySql programiranje i administracija

DML

Osnovne naredbe za pretragu

(za pracenje neophodna baza sakila)

- **SELECT**

- Select naredba je naredba koja emituje tabelarni rezultat na osnovu unetih parametara:

SELECT 'hello'

- Iz primera možemo zaključiti da SELECT, iako se često koristi za prikazivanje podataka, nije zadužena za logiku preuzimanja tih podataka, već samo za njihov prikaz.

Preuzimanje podataka iz tabele

- SELECT može takođe posedovati i treći deo, a to je izvor podataka. Preuzimanje rezultata iz izvora podrazumeva i ključnu reč from, kojoj sledi izvor (tabela):

SELECT * FROM film

- Ovaj upit vraća sve redove i sve kolone tabele film

Restrikcija kolona

- Često, nećemo biti zainteresovani za prikaz svih kolona jedne tabele. Jer, ukoliko unesemo upit:

SELECT * FROM actor

dobijamo podatke iz svih kolona o svim glumcima upisanim u tabelu actor

- Ukoliko bismo želeli da dobijemo samo imena svih glumaca u bazi, napisali bismo:

SELECT first_name FROM actor

- Ukoliko bismo želeli da prikazemo i imena i prezimena, napisali bismo:

SELECT first_name, last_name FROM actor

- Još jednom da pomenem savet za dobavljanje podataka iz baze: uvek dobavljati podatke samo iz onih kolona čiji podaci će zapravo biti potrebni. Na taj način se rasterećuje baza podataka, klijentski program, kao i sama mreža kroz koju putuju podaci.

Utvrdjivanje broja unosa

- Često se može javiti potreba da utvrdite ukupan broj zapisa u nekoj tabeli. Jednostavno, nisu Vam potrebni nikakvi detalji, već samo ukupan broj zapisa, tj. redova ili linija jedne tabele.
- Ovo se postiže sledećom naredbom:

SELECT count(film_id) FROM film

- U navedenom upitu može se koristiti naziv bilo koje kolone iz tabele film umesto korišćene film_id kolone. Može se koristiti i zvezdica (*) za sve kolone. Rezultat je uvek isti.

Utvrdjivanje broja jedinstvenih zapisa

- Ukoliko je potrebno da utvrdimo koliko određenih jedinstvenih zapisa postoji, potrebno je koristiti ključnu reč DISTINCT. Na primer, želimo da znamo koliko u bazi ima glumaca koji imaju jedinstven id broj:

SELECT COUNT(DISTINCT actor_id) FROM actor

- Prethodni primer nije efikasan, ali ako napišemo nešto ovako:

SELECT COUNT(DISTINCT first_name) FROM actor

- Na ovaj način utvrđujemo koliko glumaca postoji u bazi koji imaju jedinstveno ime. Naravno, više glumaca može imati identično ime, pa će broj glumaca sa jedinstvenim imenom biti manji od ukupnog broja glumaca u bazi.

Limitiranje rezultata

- Da bismo limitirali broj rezultata koje želimo da nam baza podataka vrati, koristimo ključnu reč LIMIT:
SELECT first_name, last_name FROM actor LIMIT 2
- Naredba LIMIT može se koristiti i sa definisanjem takozvanog offset. Pomoću offset se definiše od kojeg unosa će baza započeti pretragu. Ako napišemo:
SELECT first_name, last_name FROM actor LIMIT 2, 2
- Ukoliko napišemo:
SELECT first_name, last_name FROM actor LIMIT 10, 5
- biće preskočeno prvih 10 unosa, i vraćeno narednih 5.

Stranice

- Od MySQL edicije 4.0 može se koristiti specijalna opcija **SQL_CALC_FOUND_ROWS** koja se koristi u paru sa funkcijom **FOUND_ROWS** i pomaže prilikom straniciranja
- Prvo se napiše upit za selektovanje dela zapisa, ali korišćenjem opcije **SQL_CALC_FOUND_ROWS**:

```
SELECT SQL_CALC_FOUND_ROWS first_name, last_name  
FROM actor LIMIT 10;
```

- Zatim se, jednostavno, pozove funkcija **FOUND_ROWS**:
SELECT FOUND_ROWS();

Sortiranje rezultata

- Osnovno sortiranje, obavlja se parametrom **ORDER BY**
- Ukoliko bismo hteli da dobijemo imena i prezimena svih glumaca, a uz to i još sortirana po imenu od A do Z, napisali bismo ovako:
- **SELECT first_name, last_name FROM actor ORDER BY first_name;**
- Podrazumevano sortiranje, prilikom upotrebe ključne reči ORDER BY, je sortiranje od manje vrednosti ka većoj i, ukoliko želimo da to sortiranje bude obrnuto (od veće vrednosti ka manjoj), moramo to eksplicitno naglasiti. Za to koristimo ključne reči **DESC** (od većeg ka manjem) i **ASC** (od manjeg ka većem).
- **SELECT first_name, last_name FROM actor ORDER BY first_name DESC;**

Selektovanje specifičnih zapisa (WHERE)

- Najčešće nam neće biti potrebni svi zapisi iz neke tabele, pa čak ni deo njih, što smo postizali upotrebom limitiranja. Nekada će nam biti potrebni zapis(i) koji će zadovoljiti tačno definisane kriterijume. Ovi kriterijumi navode se ključnom rečju WHERE.
- Na primer, želimo da prikazemo nazive svih filmova iz tabele film, koji imaju vrednost kolone rating PG (Parental Guidance). To ćemo postići na sledeći način:

SELECT title FROM film WHERE rating='PG';

- Za poređenje se ne mora koristiti operator jednakosti. Na primer, ukoliko želimo da prikazemo imena svih filmova koji su duži od 2 sata, napisali bismo:

SELECT title FROM film WHERE length>120;

Priblizno podudaranje teksta (LIKE)

- Ukoliko je potrebno da izvršimo pretragu na osnovu samo dela nekog podatka, možemo koristiti ključnu reč LIKE. Na primer želimo da prikazemo sve glumce sa imenom koje počinje karakterima an:

```
SELECT first_name, last_name FROM actor WHERE first_name LIKE 'an%';
```

- Na sličan način možemo da utvrdimo da li se zapis neke kolone završava na određeni način:

```
SELECT first_name, last_name FROM actor WHERE first_name LIKE '%en';
```

- Možemo utvrditi i da li zapis neke kolone uopšte sadrži neku sekvencu karaktera bilo gde (na početku, na kraju, u sredini):

```
SELECT first_name, last_name FROM actor WHERE first_name LIKE '%en%';
```

Povezivanje filtera logickim operatorima

- Često ćemo, u uslovnom izrazu, koristiti više od jednog uslova, povezanih logičkim operatorima. Na primer, ako bismo želeli da vidimo sve filmove čiji naziv počinje slovom b i čije je trajanje duže od 3 sata, napisali bismo ovako:
- **SELECT title, length FROM film WHERE title LIKE 'B%' AND length > 180;**
- U uslovni izraz čak možemo staviti i sam upit:
- **SELECT title, length FROM film WHERE title LIKE 'B%' AND length > (SELECT 180);**

Grupisanje rezultata

- SQL omogućava sintaksu po kojoj je moguće vršiti grupisanje rezultata po određenim kolonama. To se postiže upotrebom ključne reči GROUP BY. Ipak, ako bismo ovu naredbu upotrebili na sledeći način:

```
SELECT rating FROM sakila.film  
GROUP BY rating;
```

- Naredba GROUP BY, gotovo uvek se koristi sa **agregatnim funkcijama** (funkcije će biti detaljno objašnjene u narednom modulu)
- Evo, na primer, kako bi prethodni primer imao mnogo više smisla:

```
SELECT rating, count(title) FROM sakila.film  
GROUP BY rating;
```

Filtracija nakon grupisanja (HAVING)

- Da bi se izvršila filtracija rezultata nakon grupisanja, koristi se ključna reč HAVING
- Na primer:

```
SELECT rating, count(title) FROM sakila.film  
GROUP BY rating HAVING count(title) > 200;
```

Vezba

- Potrebno je selektovati sve nazive filmova koji imaju rental_rate veći od 3, a uz to su duži od 2, ali kraći od tri sata.

Resenje

- Potrebno je selektovati sve nazive filmova koji imaju rental_rate veći od 3, a uz to su duži od 2, ali kraći od tri sata.

SELECT title FROM film WHERE rental_rate>3 AND length>120 AND length<180

Vezba

- Potrebno je preuzeti sve podatke, o svim filmovima koji u svom nazivu sadrže reč bird, a uz to imaju rental_duration veći od 5.

Resenje

- Potrebno je preuzeti sve podatke, o svim filmovima koji u svom nazivu sadrže reč bird, a uz to imaju rental_duration veći od 5.

```
SELECT * FROM sakila.film WHERE title LIKE  
'%bird%' AND rental_duration>5
```

Vezba

- Potrebno je utvrditi koliko različitih rejtinga filmova postoji. Reč je o Parental Guidance rejtingu, koji je definisan u koloni rating

Resenje

- Potrebno je utvrditi koliko različitih rejtinga filmova postoji. Reč je o Parental Guidance rejtingu, koji je definisan u koloni rating.

SELECT COUNT(DISTINCT rating) FROM film